Cache Coherency Verification for CMP Based on the PSS

Yang Yang
Productivity & Efficiency vs. Ability

- Portable Stimulus
  - System and Subsystem Level Verification, Module Integration and Performance Become the Focus of Verification
- UVM/VMM/OVM
  - IP-level Verification Has Obvious Advantages
- System Verilog/CRV&MDV
- Can Design/Can not Verify
- Directed Testing

How to Improve Productivity Efficient
PSS / SLN / Perspec

Early Scenario Portable
Scenario Language
Specification
Library scenarios for the ARM architecture

PSS parsing tool
EX: Mentor InFact

Language syntax standard
based on scene description
Reuse SLN syntax rules
PSS / SLN / Perspec
PSS / SLN / Perspec
PSS / SLN / Perspec
PSS / SLN / Perspec
Cache Coherency

- **CORE0**: 32KB L1 Cache
- **CORE1**: 32KB L1 Cache
- **CORE2**: 32KB L1 Cache
- **CORE3**: 32KB L1 Cache

**32KB L1 Cache**

- **CORE0**: 256KB L2 Cache
- **CORE1**: 256KB L2 Cache
- **CORE2**: 256KB L2 Cache
- **CORE3**: 256KB L2 Cache

**256KB L2 Cache**

- **CORE0**: 8MB L3 Cache
- **CORE1**: 8MB L3 Cache
- **CORE2**: 8MB L3 Cache
- **CORE3**: 8MB L3 Cache

**8MB L3 Cache**

- **CORE0**: 16GB DDR
- **CORE1**: 16GB DDR
- **CORE2**: 16GB DDR
- **CORE3**: 16GB DDR

**16GB DDR**

- **CORE0**: 40Cycles
- **CORE1**: 40Cycles
- **CORE2**: 40Cycles
- **CORE3**: 40Cycles

**40Cycles**

- **CORE0**: 120Cycles
- **CORE1**: 120Cycles
- **CORE2**: 120Cycles
- **CORE3**: 120Cycles

**120Cycles**

- **CORE0**: 360Cycles
- **CORE1**: 360Cycles
- **CORE2**: 360Cycles
- **CORE3**: 360Cycles

**360Cycles**

- **CORE0**: 360Cycles + 57ns
- **CORE1**: 360Cycles + 57ns
- **CORE2**: 360Cycles + 57ns
- **CORE3**: 360Cycles + 57ns

**360Cycles + 57ns**
Cache Coherency

If Core1 has acquired ownership and starts updating X, the Cache Line corresponding to Core2 needs to be set to state 1.

The thread on Core1 wants to update the value of variable X.

The Cache Line corresponding to Core1 must be set to state 1 if Core2 has acquired ownership and starts updating Y.

The thread on Core2 wants to update the value of the variable Y.
Verification Model

State enumeration verification
Operate * K
State * N
Core * M

\[ C = K \times N^M \]

\[ 4 \times 5^8 = 1,562,500 \]
Verification Model

... Operate * K

State * N

Core * M
Verification Model

Cache Level
- L1
- L2
- L3

Cache Ability
- Device_Mem
- Non-Cacheable
- Cacheable

Cache Share
- Non-Shareable
- Outer-Shareable
- Inner-Shareable

Cache Replacement
- Random
- Prandom
- LRU

Cache Protocol
- None
- MOESI
- MESI
Verification Model

- Consider internal state transitions
- Verify protocol details
- Methodological verification applies

System level

- Internal state not considered
- Verify the overall Cache behavior
- Application-level models

Applied level

As systems become more complex, validation at this level becomes more difficult, and structural changes result in poor reusability.

It can flexibly use various application-level models to find logical bugs faster, and it does not affect its reusability when the structure changes and the behavior remains unchanged.
Verification Model

State Diagram

Scenario

Test Case
Verification Model

Core Access Cache Race Model

$4 \times 4 \times 13 = 208$
Plan

False Sharing
- Min False Sharing coherency
- Multi False Sharing coherency
- Max False Sharing coherency

True Sharing
- Min True Sharing coherency
- Multi True Sharing coherency
- Max True Sharing coherency

Cache State
- Debug Cadence Scenarios

Analyze
- Research New Model to Cover State Transition
- Symbol state model method …..
Thank You!

linshuying1994@163.com